

Highlights

Every pair decided: the nineteen kissing-number configurations of an autonomous discovery system, certified exactly from their published bytes*

Carlos Toledo

- All 19 kissing-number bounds of the Qiushi Engine hold: 9.75×10^{15} pairs decided exactly.
- The exactness argument is one inequality per path, checked against the data and asserted before the run.
- Seven configurations of about 1.4×10^{15} pairs each decided on one GPU for \$8.15.
- A refusal with the missing files named was answered within days; the claimant linked the check.
- What is decided is each configuration, not optimality and not the record in its dimension.

Every pair decided: the nineteen kissing-number
configurations of an autonomous discovery system,
certified exactly from their published bytes

Carlos Toledo^{a,*}

^aIndependent researcher, the cert-machine project, Brazil

Abstract

In September 2026 an autonomous multi-agent research system published new lower bounds for the kissing number in 19 dimensions (25, 27, 32–39, 43, 45 and 49–55), each a configuration of integer or closed-form points, each proved by moment identities and structure, and each released with a verification file that records that the pairwise inner products were not checked. We decided every pair. A kissing configuration is the simplest universal claim there is—a finite set of points and one inequality per pair—and with integer coordinates the inequality is exact: for rows of squared norms A and B with inner product p , the angle is at least sixty degrees if and only if $p \leq 0$ or $4p^2 \leq AB$. The products were formed in floating-point hardware and are exact because every partial sum is bounded by the Cauchy–Schwarz inequality below the integer range the format holds, a bound checked against the data before anything ran and now asserted in the code; in dimensions 25 and 27, whose coordinates are integer combinations of 1, $\sqrt{2}$, $\sqrt{3}$ and $\sqrt{6}$, the pairs a floating-point screen could not separate were decided in arbitrary-precision integers over the field. All 19 configurations hold: 9.75×10^{15} pairs decided, 6.72×10^{12} of them at exactly sixty degrees and none closer; every count matches the published one and every configuration whose span is recorded spans exactly d dimensions. The seven largest, about 1.4×10^{15} pairs each, ran on one rented NVIDIA L40S for 6.8 GPU-hours after a calibration that reproduced the laptop’s exact contact counts in two smaller dimensions; a seeded sample of every configuration was re-decided in plain integers by a program that shares no code with the bulk path. The claimant published its points within days of the request and linked the check beside its results table. A configuration that holds proves a lower bound and nothing

*Draft v0.1 of 2026-10-21, a companion to the cert-machine method paper, built from repository commit 16849460e608. Every measured number and count is a macro read at build time from the records named in the Data availability statement; a record that no longer supports a sentence stops the build.

*Corresponding author.

Email address: `carlos@carlostoledo.co` (Carlos Toledo)

more; the field’s table lists higher values in dimensions 25, 27 and 33, from constructions whose points have not been published.

Keywords: kissing numbers, spherical codes, machine-generated mathematics, independent verification, exact integer arithmetic, tensor cores, certificates

1. Introduction

The kissing number $K(d)$ is the largest number of non-overlapping unit spheres that can touch a central unit sphere in $\mathbb{R}^d$. It is known exactly in dimensions 1 to 4, 8 and 24 (Leech, 1967; Odlyzko and Sloane, 1979; Musin, 2008; Conway and Sloane, 1999); elsewhere it is bracketed by constructions below and by linear and semidefinite programming above (Bachoc and Vallentin, 2008; Mittelmann and Vallentin, 2010; Leijenhorst and de Laat, 2024), and the lower bounds have moved more in the last two years than in the preceding four decades. Cohn and Li raised dimensions 17 through 21 by altering signs in Leech-lattice vectors (Cohn and Li, 2024), and a binary code raised dimension 19 again (Ho, 2026); Ganzhinov’s highly symmetric line systems set records in dimensions 10, 11 and 14 (Ganzhinov, 2022); a language-model-guided program search found a 593-point configuration in dimension 11 (Novikov et al., 2025; Georgiev et al., 2025), and that dimension then moved twice more, each time by a machine, to 604 (Toledo, 2026a, the dimension-eleven paper). In June 2026 a numerical 841-point arrangement was reported in dimension 12 (Takhanov et al., 2026), and in September three parties published bounds between dimensions 18 and 55 within weeks of each other (Kravatskiy, 2026; Takhanov and Yun, 2026; Yang et al., 2026). In high dimensions the asymptotic lower bound was improved by a factor linear in d (Jenssen et al., 2018; Fernández et al., 2021).

The largest of the September releases is the one this paper decides: the Qiushi Engine, an autonomous multi-agent research system, published 19 new lower bounds in dimensions 25, 27, 32–39, 43, 45 and 49–55 (Yang et al., 2026). Each bound is a configuration built from lattices, codes, sections and projections; its size is proved by moment identities and the angular condition by structure; and the release’s own verification files record, for every configuration they cover, that the pairwise inner products were not checked. That sentence made this exercise possible: whether every pair of a finite set of points satisfies one inequality is a finite computation that a party with no stake in the result can make from the published bytes.

We made it. Every pair of every configuration was decided exactly, by programs that share no code with the claimant, from files pinned by the digests the claimant’s manifest and the hosting platform report. Two configurations were published with the paper as exact objects and were

decided at once; the other seventeen had no published points and were recorded as NEEDS DATA with the missing files named, a refusal that the claimant answered within days by publishing the points, in exactly the form the check needed. Every one of the 19 bounds holds. The contribution of this paper is not a bound—no bound is improved—but the decision: what was read, how each pair was decided exactly in floating-point hardware, what the second implementation and the calibration do and do not establish, what it cost, and what the claimant did next. It is a companion to the cert-machine method paper (Toledo, 2026a, the method paper), whose grammar of three verdicts with witnesses it applies to one kind of claim; the public register holds its rows (Toledo, 2026b). Section 2 gives the materials, Section 3 the method, Section 4 the results, Section 5 the discussion and the limits, and Section 6 concludes; Appendix A lists the command that re-derives each result.

2. Materials

The claim and its bytes. The paper (Yang et al., 2026) (arXiv:2609.35051 (v1 2026-09-28)) states the 19 bounds of Table 1; the bound each row improves on is the one the claimant cites as previous, and the gains run from 1 (dimensions 25 and 27) to 8,736. The claimant’s repository (Oxelra-AI, 2026) was pinned at commit `f3060ec86137` of 2026-09-29 (code MIT; reports, notes and original data CC BY 4.0 (LICENSE, RIGHTS.md); the inherited Kravatsky data under its MIT notice (constructions/d25/THIRD_PARTY_LICENSE.txt)). It holds two constructions as exact objects: the dimension-25 configuration, 8 families of points of which 88 are rational caps and repair points stated in closed form, and the dimension-27 configuration, 4 families. Both inherit the first layer of the dimension-25 and -27 configurations of Kravatskiy (2026), whose data the repository ships under its own notice, and add the points that raise each bound by one; the record checks that inheritance fact by fact (the owners, the leans, the rational heads and the extra lean of the dimension-25 construction equal Kravatskiy’s files; the first layer of the dimension-27 construction is his byte for byte). The other 17 configurations were published after our request as three releases of the repository, `configuration-vectors-2026-10-12` (dimensions 32, 33, 34, 35, 36, 37, 38 and 39), `lifting-vectors-2026-10-12` (dimensions 49, 50, 51, 52, 53, 54 and 55) and `section-vectors-2026-10-11` (dimensions 43 and 45): the sections of dimensions 43 and 45 as 48-column integer rows, the others as signed 8-bit integer rows, except dimension 38, whose 1,932 rotated-axis rows are 64-bit integers. Each file is pinned by two digests: the one the hosting platform reports for the archive and the one the claimant’s verification file records for the decompressed rows. A file that does not match either is refused before it is read.

The machines. The 12 smaller configurations were decided on an 8-core laptop with 16 GB of memory: Python 3.9.6, numpy 2.0.2 on Accelerate (numpy build config), one BLAS thread per worker. The 7 largest, about 1.4×10^{15} pairs each, ran on one rented NVIDIA L40S (46,068 MiB, driver 580.126.09; torch 2.4.1+cu124, cuda 12.4, numpy 1.26.3, triton (bundled with torch)) at \$1.09 per hour, created 2026-10-12 13:03 UTC and terminated once its result files were read. The pod held no credential: it fetched the deciding script from this repository at a fixed commit (6371ba3d7e28), downloaded each configuration from the claimant’s public release, refused any file that did not match the pins written into that script, and served its results for download.

The record. Every decision writes a row of `certs/kissing-wave.json`, the record of the September wave (79 rows from 3 claimants, of which the 19 here): the inputs by path and digest, the counts, the verdict, the witness, the machine, the time, and the second implementation’s reading. Every number in this paper is a macro written by a generator from that record and the pod’s files, and the generator refuses to write when a record no longer supports the sentence that quotes it.

3. Method

3.1. The reading

In the grammar of the method paper a claim is a pair of a domain and a predicate, read universally or epistemically; a kissing configuration is the universal case at its simplest. The domain is the finite set of pairs of distinct rows; the predicate is $\langle x, y \rangle \leq \frac{1}{2}|x||y|$. With integer coordinates the predicate is exact. For rows of squared norms A and B with inner product p ,

$$\angle(x, y) \geq 60^\circ \iff p \leq 0 \text{ or } 4p^2 \leq AB, \quad (1)$$

and a contact, at exactly sixty degrees, is $4p^2 = AB$ with $p > 0$. The same test rejects a repeated or parallel row, since then $p^2 = AB$ with $p > 0$, so distinctness of the directions is part of the pair test; a zero row is rejected separately, by its norm. The rows are first grouped by squared norm, the counts per norm compared with the claimant’s, and the verdict is WITNESSED if every pair satisfies (1) and every count matches, UNWITNESSED otherwise, with what failed; the row count, the norms and the span are facts recorded beside the verdict.

3.2. Three arithmetics, each exact on its data

The work is the pairs, and the pairs are integer products. Three paths form them, and each is exact for a reason stated once, checked against the data before the path runs and, since 2026-10-21, asserted in the code so that a configuration outside the bound cannot be decided by that path at all.

Single-precision BLAS on the laptop. A product of integer vectors is an integer, and floating-point hardware computes it exactly when no partial sum can leave the range the format holds exactly— 2^{24} for single precision. By the Cauchy–Schwarz inequality every partial sum of products is at most $|x||y|$, at most the largest squared norm in the configuration, which is 4,608 at most (dimension 25) on the laptop. The tiles are therefore exact integers, and the test (1) is applied to them as integers: $p \leq t(A, B) = \lfloor \sqrt{AB}/2 \rfloor$, with the threshold computed by exact integer square root. The script asserts the bound below 2^{24} before the float32 path runs.

Dimension 38. Its 1,932 rotated-axis rows have squared norm 3.95×10^{35} , a 36-digit integer, beside 48 for the other rows, so three parts are decided three ways: the small rows against each other in float32 BLAS under the bound 48 (174,030,825,528 pairs); the small rows against the axes in 64-bit integers, every partial sum bounded by $\sqrt{48N} < 2^{63}$ (1,139,818,176 pairs, no contact); and the axes against each other in Python’s arbitrary-precision integers (1,865,346 pairs, 159,264 contacts).

A fused GPU kernel. The configurations of dimensions 49 to 55 are signed 8-bit rows of 51 to 72 integer columns with squared norms 192, 144, 48 and coordinates at most 12 in size. A Triton kernel casts the rows to half precision in registers, forms the products on the tensor cores with single-precision accumulation, and reduces each 128×128 block to its largest inner product and its counts; the host applies (1) per pair of norm blocks. The argument asks of the hardware only that every product and every partial sum, each an integer of magnitude at most 192, be held exactly at every stage. Half precision holds every integer below 2^{11} exactly and single precision every integer below 2^{24} (IEEE, 2019); the block fused multiply-add that tensor cores implement forms the products exactly and accumulates them in the wider format (Blanchard et al., 2020), and the alignment and truncation of partial products that Fasi et al. (2021) measured on Volta, Turing and Ampere devices, and that a formal model of those three generations reproduces (Valpey et al., 2025), discard bits only below the unit of the largest aligned operand, which for integers of this size is never a bit of an integer. The script asserts the bound below 2^{11} before any GPU path runs. What the argument does not cover is stated in Section 5.

Closed forms in dimensions 25 and 27. The claimant’s two constructions state their points in closed form, as combinations such as $\sqrt{3}/2 + \sqrt{2}/4$; after scaling, every coordinate is an integer combination of 1, $\sqrt{2}$, $\sqrt{3}$ and $\sqrt{6}$, and so is every inner product. The engine screens the pairs in single-precision tiles under the same Cauchy–Schwarz bound and decides every pair the screen cannot separate by a proved margin—1,118,436 pairs in dimension 25—in arbitrary-precision integers over $\mathbb{Q}(\sqrt{2}, \sqrt{3})$, where the sign of $4p^2 - AB$ is decided exactly and a tie raises rather than returns a sign (a red control forces one). The 88 rational caps and repair points of dimension 25 are decided by the exact path throughout.

3.3. *The span, and the counts*

That the rows span exactly d dimensions is decided by the rank over $\mathbb{Q}$ of the exact Gram matrix $X^T X$, summed in 64-bit integer chunks, which equals the rank of X . For the sections of dimensions 43 and 45—48-column rows of squared norm 576 and 4,416—the rank of a sample is computed by exact rational elimination and every row is shown orthogonal to the 5 and 3 integer vectors spanning the sample’s null space; in dimensions 25 and 27 the span is not recorded. For every configuration the row count and, where the claimant states them, the counts per squared norm are compared with the claimant’s.

3.4. *The sections decided a second time, in the form their construction states*

Dimensions 43 and 45 were decided twice. Once in Cartesian form, as above. Once in Gram form, from the claimant’s published lattice data, read as data and never run: the Gram matrix G of the parent lattice, its integer basis B with $BB^T = 96G$ for dimension 43, and the section conditions the construction defines. Every Cartesian row is written as an integer combination of the basis (a rounded solve, then the identity checked exactly for every row), the combination is shown to satisfy the section conditions, and every pair is decided in Gram form; the counts—pairs, contacts, the nearest non-contact as a cosine—must equal the Cartesian record’s. They do. This is a second reading of the same bytes in the claimant’s own terms, and it checks that the points are the set the construction defines, which the Cartesian decision alone does not.

3.5. *The second implementation*

Every bulk path is checked by a program that shares no code with it. A seeded sample of 600 to 726 rows of each configuration, together with every row of the blocks of at most two hundred rows, has all its pairs decided in plain Python integers with the test (1) written out—no numpy, no BLAS, no GPU—and its contacts and violations must equal the bulk path’s counts on the same rows. The sample is written into the record and decided a third time by a JavaScript reading over arbitrary-precision integers, the rows as strings, on the laptop. For the two closed-form constructions the JavaScript sample is decided by a separate basis reading; in dimension 25 that reading excludes the families rational caps and repair points, so its 664 rows and 220,116 pairs leave out the rows that make the bound new, and the Python engine’s own sample of 13,194 pairs is the only second reading that covers them. These are N-version checks: written by the same author, with AI agents, from one specification, they catch a slip in one arithmetic and cannot catch a misreading the two share (Knight and Leveson, 1986), and they decide samples, not the configurations.

3.6. Calibrate before you decide

A new path decides nothing until it has reproduced a decision already made by an old one. The wave runner re-derives $K(8) = 240$ from the roots of E_8 and $K(24) = 196,560$ from the whole Leech shell, every pair, at the start of every run (2 calibrations), and its 5 red controls—a sign flipped, a point rotated by thirty degrees, a row listed twice, a coordinate moved off the sphere, a forced tie—must each be caught before a row is written. The GPU kernel was required to return the laptop’s exact contact counts, 358,430,976 and 325,490,632, on the configurations of dimensions 32 and 39 (1.4 and 3 seconds) before it was allowed to decide the seven large ones; and the pre-registration of the paid run stated beforehand what would be measured, what counted as a result, and at what cost the run stopped.

4. Results

The verdict. WITNESSED in all 19 dimensions (Table 1, Figure 1): 9.75×10^{15} pairs decided, 6.72×10^{12} of them at exactly sixty degrees and none closer; every count matches the published one, every configuration whose span is recorded spans exactly d dimensions, every seeded sample agrees with its bulk path in plain integers and in the JavaScript reading, and the two sections agree with their Gram-form re-decision. In dimension 45 the nearest non-contact pair has cosine 0.478. Every bound in the paper holds.

d	previous, as cited	points claimed	pairs decided	at exactly 60°	arithmetic	time
25	197,579	197,580	19,518,829,410	446,074,805	float32 + exact field	37 s
27	201,566	201,567	20,314,526,961	444,139,680	float32 BLAS	27 s
32	346,944	347,584	60,407,144,736	358,430,976	float32 BLAS	45 s
33	362,048	363,968	66,236,170,528	368,310,912	float32 BLAS	52 s
34	381,124	384,196	73,803,091,110	380,481,792	float32 BLAS	59 s
35	409,548	409,676	83,917,007,650	315,579,800	float32 BLAS	1 min
36	484,568	484,760	117,495,886,420	404,388,832	float32 BLAS	1 min
37	496,232	498,024	124,013,703,276	392,397,168	float32 BLAS	1 min
38	591,612	591,900	175,172,509,050	379,853,270	float32, int64, integers	3 min
39	756,116	763,668	291,594,025,278	325,490,632	float32 BLAS	3 min
43	2,545,056	2,553,792	3,260,925,512,736	4,467,331,098	float32 BLAS	40 min
45	7,379,838	7,380,090	27,232,860,514,005	19,832,543,736	float32 BLAS	9.1 h
49	52,430,140	52,430,156	1,374,460,602,877,090	965,197,554,294	fused GPU kernel	53 min
50	52,458,418	52,458,468	1,375,945,406,224,278	964,169,164,800	fused GPU kernel	53 min
51	52,500,816	52,500,930	1,378,173,799,181,985	962,629,120,340	fused GPU kernel	53 min
52	52,585,516	52,585,772	1,382,631,682,125,106	959,560,825,568	fused GPU kernel	53 min
53	52,698,222	52,698,696	1,388,576,253,700,860	955,491,412,444	fused GPU kernel	53 min
54	52,922,906	52,923,774	1,400,462,900,739,651	947,433,469,328	fused GPU kernel	1.2 h
55	53,299,730	53,301,140	1,420,505,735,999,230	934,069,080,624	fused GPU kernel	1.2 h

Table 1: The 19 configurations of Yang et al. (2026): the bound the claimant cites as previous, the bound claimed, and every pair decided (`certs/kissing-wave.json`). Times are wall-clock: the laptop’s 8 cores for the BLAS rows (the dimension-45 run, 9.1 h, includes a pause), one NVIDIA L40S for the fused kernel. In all, 9,752,282,639,769,360 pairs and 6,716,665,650,099 contacts; no pair closer than sixty degrees.

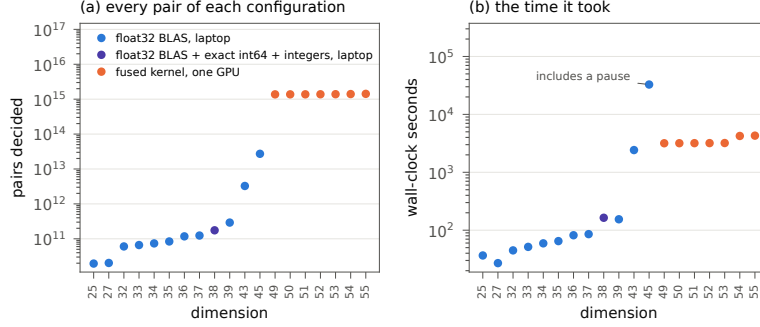


Figure 1: The 19 configurations: (a) the pairs decided in each dimension and (b) the wall-clock time, by the arithmetic used (`certs/kissing-wave.json`). The seven largest ran on one GPU; the dimension-45 point includes a pause.

The cost. The 12 laptop rows took 35,954 seconds of wall-clock in all; the 7 GPU rows 6.8 GPU-hours of deciding at \$1.09 per hour, and the pod’s whole life, its calibration and about 1.3 h idle after it finished included, was billed \$8.15 as read at 21:25 UTC on 2026-10-12 (`corpus/kissing/wave/qiushi-gpu-2026-10-12/pod.json`). The exactness argument is one inequality per path, checked against the data before the run; nothing in the cost is spent on rigour.

The refusal, and how it closed. The two constructions were decided at once. The other seventeen rows were NEEDS DATA from the request of 2026-10-06, on the repository’s public thread (<https://github.com/Oxelra-AI/Qiushi-Engine-Kissing-Number-Research/issues/1>), until the points were released on 2026-10-11 and 2026-10-12: each row named the files that would decide it. The claimant published the points for dimensions 43 and 45 five days after the request and, once those were decided, every remaining configuration within hours (Oxelra-AI, 2026). A refusal that names what would convert it is a result with a size, and here the size was a few days.

What the claimant did next. It published its points in exactly the form the check needed, and it linked the check from both of its READMEs beside its results table (pinned at commit `bbb7aafd78e7` of 2026-10-19 in `corpus/kissing/wave/qiushi-readmes-2026-10-19`). We draw no causal conclusion from the sequence; it is reported as what happened.

The record in its dimension. A configuration that holds proves a lower bound and nothing more. The field’s table of kissing-number bounds (Cohn, 2026), read on 2026-10-11, lists higher values in dimensions 25, 27 and 33, from constructions whose points have not been published and which this system has not decided; it lists the bounds decided here in dimensions 32, 34–39, 43 and 45, and it does not cover 49–55.

5. Discussion

What the exercise shows. No bound was found false. The gap was between what was claimed and what was checked: a pairwise condition proved by structure and never checked pair by pair, and closed by a finite computation whose only hard part was deciding to make it. The claimant’s checker stopped exactly where the claim exceeded it, and said so; an independent decision is not a better checker but one that was not written by the claimant and so does not know where to stop. The grammar made three things possible that a confirmation by e-mail does not: a WITNESSED with its record can be cited, a NEEDS DATA that names its files can be answered, and the scope—each configuration, not the record in its dimension—travels with the verdict.

Relation to the literature. Independent re-decision is routine for the exact cases of the kissing problem, whose proofs are checked by reading (Musin, 2008) or by formalisation, and for upper bounds, where the semidefinite programs of Bachoc and Vallentin (2008) are solved to high accuracy with the dual solution serving as a certificate (Mittelmann and Vallentin, 2010; Leijenhorst and de Laat, 2024). For lower bounds the certificate is the configuration itself, and the field’s convention has been to trust the construction’s proof of the angular condition. The machine-generated releases of 2025 and 2026 (Novikov et al., 2025; Yang et al., 2026) changed the size of the objects and the way they are proved—by search and by structure, with the pairwise check left to the reader—and the exercise here is the reader’s check made explicit, at a cost measured in hours and a few dollars. The exactness of integer products in floating-point hardware under a Cauchy–Schwarz bound is elementary; what the tensor-core literature supplies is the characterisation of what the hardware does with aligned partial products (Fasi et al., 2021; Blanchard et al., 2020; Fasi et al., 2023; Valpey et al., 2025; Khattak and Mikaitis, 2025), which the argument of Section 3.2 needs only in its weakest form.

Threats to validity.

- *Selection.* The release was chosen because it had finite objects to decide and had said that it had not decided them. The result describes these 19 configurations, not machine-generated mathematics at large.
- *The author in the outcomes.* The points were published and the check linked after a request the author made. They are reported as what happened, and no causal claim rests on them.
- *The second implementation.* The plain-integer and JavaScript readings are N-version checks by the same author from one specification, on samples; in dimension 25 the JavaScript sample leaves out the rows

that make the bound new. They are counted apart from a re-derivation by another party, of which there is none.

- *The GPU.* Fasi et al. (2021) characterised Volta, Turing and Ampere tensor cores, and the formal model of Valpey et al. (2025) covers the same three generations; the pod’s NVIDIA L40S belongs to a later one, which an architecture-independent test scheme has since been applied to (Khattak and Mikaitis, 2025), but not by us. The exactness argument for the seven GPU rows therefore rests on the documented single-precision accumulation of the block fused multiply-add, on the assertion that every partial sum is an integer below 2^{11} , and on the calibration that reproduced the laptop’s exact contact counts in dimensions 32 and 39 on this device. The calibration is a check, not a proof: a device that held integers below 2^{11} inexactly would fail it, and one that failed only on inputs the calibration did not exercise would not. Re-deciding dimensions 49 to 55 with integer matrix multiplication, or characterising the device, would close the gap; neither has been done.
- *The span of 25 and 27.* It is not recorded; the row counts and the pair test are.

What is not claimed. A configuration that holds proves a lower bound. It says nothing about optimality, about any upper bound, about the record in its dimension, or about the other steps of the discovery process that produced it; the moment identities and the structural proofs of the paper were not read for this exercise and are not decided by it. “Independent” means independent of the claimant: no program of the claimant’s was run, and the lattice data of Section 3.4 were read as data. Nothing is a probability.

6. Conclusions and recommendations

Every one of the 19 kissing-number bounds of Yang et al. (2026) holds, pair by pair: 9.75×10^{15} pairs decided exactly from the published bytes, 6.72×10^{12} contacts and no violation, for 35,954 seconds of a laptop and \$8.15 of one GPU. The decision was possible because the claimant published its points and said what it had not checked, and it was cheap because exact integer arithmetic in floating-point hardware costs nothing beyond one inequality per path.

To producers of machine-generated constructions: publish the object, not only its description—the points in one integer form, with a digest beside each file—because the finite check is cheap once the bytes exist and impossible until they do; and state the scope of every check in the release itself. “Pairwise inner products checked: false” is the right sentence, and it was the one that made this paper possible. To those who maintain the

tables of a field: distinguish a bound whose configuration has been decided from outside from one that has only been constructed, and say what moves a bound from the second class to the first.

Declaration of competing interest

The author declares no competing financial or personal interests. The Qiushi Engine group has not been consulted on this draft; the exchange with the group took place on the public thread cited, and its READMEs link the check described here. Prospective co-authors are reviewing this manuscript.

CRediT authorship contribution statement

Carlos Toledo: conceptualization, methodology, software, validation, investigation, data curation, writing—original draft, writing—review and editing. The author list and the roles are to be agreed with the prospective co-authors.

Data availability

Every record, script and table source is public in the cert-machine repository (<https://github.com/carlostoledo1891/cert-machine>, commit 16849460e608; archived as Toledo, 2026a): the decisions `certs/kissing-wave.json`; the GPU pod’s results, calibration, build log and bill `corpus/kissing/wave/qiushi-gpu-2026-10-12/`; the pinned READMEs `corpus/kissing/wave/qiushi-readmes-2026-10-19/`; the wave manifest `corpus/kissing/wave.meta.json`; the claimant’s lattice data `corpus/sources/kissing/`; the field’s table as read `corpus/sources/kissing/cohn-table-extract_2026-10-11.json`; the runners `tools/run-kissing-wave.py`, `tools/run-kissing-configs.py`, `tools/run-kissing-config38.py`, `tools/run-kissing-sections.py`, `tools/run-kissing-gram.py` and `tools/runpod-qiushi-lifting.sh`; and the generator of this paper’s numbers `tools/paper-numbers/qiushi-kissing.js`. The claimant’s configurations are cited at their source (Oxelra-AI, 2026) and pinned by digest in the records; they remain under the claimant’s licence.

Declaration of generative AI and AI-assisted technologies

During the preparation of this work the author used large language models (Claude, Anthropic) to draft code, exposition and this manuscript in a certified-verification setting: constructions and prose were proposed with the help of the models, and every quantitative claim is decided by outward-rounded interval arithmetic or exact rational arithmetic, with adversarial

controls that must fire for a record to be accepted. After using the tool the author reviewed and edited the content and takes full responsibility for the content of the publication. The records, not the models, are the evidence offered; they are public and re-runnable. For this paper specifically: the deciding scripts, the GPU kernel, the second implementations and the prose were drafted by AI agents (Claude) under written briefs and consolidated into the public record; every verdict here is a computation, and no reading by an agent enters any verdict.

Funding

This work received no external funding. The rented GPU (\$8.15) was paid by the author.

Appendix A. Reproduction

Each headline result re-derives with the command beside it in Table A.1, on the laptop of Section 2 unless the row says otherwise; each command that fetches a claimant’s file refuses it unless it matches the pinned digest, and each run re-derives the E_8 and Leech calibrations and fires its red controls before a row is written.

result	command	expected	time
kissing, $d = 25$, 27	<code>python3 tools/ run-kissing-wave.py --only qiushi-25,qiushi-27</code>	WITNESSED; the constructions rebuilt from the repository's files	as in Table 1
kissing, $d = 32$ –37, 39	<code>python3 tools/ run-kissing-configs.py <d></code>	WITNESSED; the pairs and contacts of Table 1	as in Table 1
kissing, $d = 38$	<code>python3 tools/ run-kissing-config38.py</code>	WITNESSED; the three parts of Section 3.2	as in Table 1
kissing, $d = 43$, 45	<code>python3 tools/ run-kissing-sections.py <d></code>	WITNESSED; the Gram form by <code>python3 tools/ run-kissing-gram.py qiushi <d></code>	as in Table 1
kissing, $d = 49$ –55	<code>tools/ runpod-qiushi-lifting.sh, on a CUDA machine; then python3 tools/ run-kissing-configs.py --import DIR</code>	the calibration on 32 and 39 first, then WITNESSED	6.8 GPU- hours
the JavaScript readings	<code>python3 tools/ run-kissing-configs.py --js <d>; python3 tools/ run-kissing-sections.py <d> --js-only</code>	the recorded sample's counts, equal to the bulk path's	seconds
this paper's numbers	<code>node tools/paper-numbers/ qiushi-kissing.js</code>	the macro file rewritten identically from the records	seconds

Table A.1: Each headline result, the command that re-derives it, what it must print, and what it took (the times are those the records hold).

References

- Bachoc, C., Vallentin, F., 2008. New upper bounds for kissing numbers from semidefinite programming. *Journal of the American Mathematical Society* 21, 909–924. doi:10.1090/S0894-0347-07-00589-9.
- Blanchard, P., Higham, N.J., Lopez, F., Mary, T., Pranesh, S., 2020. Mixed precision block fused multiply-add: error analysis and application to GPU tensor cores. *SIAM Journal on Scientific Computing* 42, C124–C141. doi:10.1137/19M1289546.
- Cohn, H., 2026. Kissing numbers. URL: <https://cohn.mit.edu/kissing-numbers>. The table of bounds, read 2026-10-11; the extract is pinned in the repository.
- Cohn, H., Li, A., 2024. Improved kissing numbers in seventeen through twenty-one dimensions. *arXiv:2411.04916*.
- Conway, J.H., Sloane, N.J.A., 1999. *Sphere Packings, Lattices and Groups*. 3rd ed., Springer, New York. doi:10.1007/978-1-4757-6568-7.

- Fasi, M., Higham, N.J., Lopez, F., Mary, T., Mikaitis, M., 2023. Matrix multiplication in multiword arithmetic: error analysis and application to GPU tensor cores. *SIAM Journal on Scientific Computing* 45, C1–C19. doi:10.1137/21M1465032.
- Fasi, M., Higham, N.J., Mikaitis, M., Pranesh, S., 2021. Numerical behavior of NVIDIA tensor cores. *PeerJ Computer Science* 7, e330. doi:10.7717/peerj-cs.330.
- Fernández, I.G., Kim, J., Liu, H., Pikhurko, O., 2021. New lower bounds on kissing numbers and spherical codes in high dimensions. *arXiv preprint arXiv:2111.01255*.
- Ganzhinov, M., 2022. Highly symmetric lines. *arXiv:2207.08266*.
- Georgiev, B., Gómez-Serrano, J., Tao, T., Wagner, A.Z., 2025. Mathematical exploration and discovery at scale. *arXiv:2511.02864*.
- Ho, B.S., 2026. A new lower bound for the kissing number in 19 dimensions. *arXiv preprint arXiv:2603.10425*.
- IEEE, 2019. IEEE standard for floating-point arithmetic. doi:10.1109/IEEESTD.2019.8766229. IEEE Std 754-2019.
- Jenssen, M., Joos, F., Perkins, W., 2018. On kissing numbers and spherical codes in high dimensions. *Advances in Mathematics* 335, 307–321. doi:10.1016/j.aim.2018.07.001.
- Khattak, F.A., Mikaitis, M., 2025. Generalized methodology for determining numerical features of hardware floating-point matrix multipliers: part I, in: 2025 IEEE High Performance Extreme Computing Conference (HPEC), IEEE. doi:10.1109/HPEC67600.2025.11196657.
- Knight, J.C., Leveson, N.G., 1986. An experimental evaluation of the assumption of independence in multiversion programming. *IEEE Transactions on Software Engineering* SE-12, 96–109. doi:10.1109/TSE.1986.6312924.
- Kravatskiy, A., 2026. KissingNumbers: configurations and verifications. GitHub repository. URL: <https://github.com/alexlegeartis/KissingNumbers>.
- Leech, J., 1967. Notes on sphere packings. *Canadian Journal of Mathematics* 19, 251–267. doi:10.4153/CJM-1967-017-0.
- Leijenhorst, N., de Laat, D., 2024. Solving clustered low-rank semidefinite programs arising from polynomial optimization. *Mathematical Programming Computation* 16, 503–534. doi:10.1007/s12532-024-00264-w.

- Mittelmann, H.D., Vallentin, F., 2010. High accuracy semidefinite programming bounds for kissing numbers. *Experimental Mathematics* 19, 174–178. [arXiv:0902.1105](#).
- Musin, O.R., 2008. The kissing number in four dimensions. *Annals of Mathematics* 168, 1–32. doi:10.4007/annals.2008.168.1.
- Novikov, A., Vű, N., Eisenberger, M., Dupont, E., Huang, P.S., Wagner, A.Z., Shirobokov, S., Kozlovskii, B., Ruiz, F.J.R., Mehrabian, A., Kumar, M.P., See, A., Chaudhuri, S., Holland, G., Davies, A., Nowozin, S., Kohli, P., Balog, M., 2025. AlphaEvolve: a coding agent for scientific and algorithmic discovery. [arXiv:2506.13131](#).
- Odlyzko, A.M., Sloane, N.J.A., 1979. New bounds on the number of unit spheres that can touch a unit sphere in n dimensions. *Journal of Combinatorial Theory, Series A* 26, 210–214. doi:10.1016/0097-3165(79)90074-8.
- Oxelra-AI, 2026. Qiushi-Engine-Kissing-Number-Research: the complete point sets. URL: <https://github.com/Oxelra-AI/Qiushi-Engine-Kissing-Number-Research>. Releases section-vectors-2026-10-11, configuration-vectors-2026-10-12 and lifting-vectors-2026-10-12, and issue #1; data CC BY 4.0.
- Takhanov, R., Assylbekov, Z., Yun, S., 2026. Structure of kissing arrangements in $\mathbb{R}^{12}$ and a place for the 841st sphere. *arXiv preprint arXiv:2606.18984*.
- Takhanov, R., Yun, S., 2026. New lower bounds for kissing numbers in dimensions 25–31. *arXiv preprint arXiv:2609.21591*.
- Toledo, C., 2026a. cert-machine: independent exact certification of machine-generated mathematics. doi:10.5281/zenodo.22225860. The repository archive: code, records and reports. The concept DOI resolves to the latest version; the version this paper was built from is named on its title page.
- Toledo, C., 2026b. Independent exact certification of machine-generated mathematics: the first register. URL: <https://carlostoledo.co/paper/register.pdf>. Archived with the repository, doi:10.5281/zenodo.22225860. Draft.
- Valpey, B., Li, X., Pai, S., Gopalakrishnan, G., 2025. An SMT formalization of mixed-precision matrix multiplication: modeling three generations of tensor cores. *arXiv preprint arXiv:2502.15999*.
- Yang, S., Zhao, R., Wu, J., Wang, Y., Chen, F., Zhu, K., Li, W., Li, Z., Li, Y., Hong, S., Pan, Y., Yang, J., Deng, T., Mi, J., Chen, H., Yang, Y.,

2026. Large-scale autonomous discovery of kissing number constructions.
[arXiv:2609.35051](#).